Attorney's Docket No.: 10559/043001/P7397/Intel Corporation

AF/2155
120

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant : Anthony Toivonen Art Unit : 2155
Serial No.: 09/430,691 Examiner : Thu Ha Nguyen
Assignee : Intel Corporation
Filed : October 29, 1999
Title : DISTRIBUTED COMPONENT SYSTEM MANAGEMENT

MAIL STOP APPEAL BRIEF - PATENTS
Commissioner for Patents
Washington, D.C. 20231

BRIEF ON APPEAL

Sir:

Applicant herewith files this brief on appeal thus perfecting the notice of appeal which was originally filed on March 10, 2004.

The headings and subject matter required by rule 192 follow.

(1) Real Party in Interest

This case is assigned of record to Intel Corporation, who is hence the real party in interest.

CERTIFICATE OF MAILING BY FIRST CLASS MAIL

I hereby certify under 37 CFR §1.8(a) that this correspondence is being deposited with the United States Postal Service as first class mail with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

June 10, 2004
Date of Deposit

R. P. Ippolito
Signature

Roxanne Ippolito
Typed or Printed Name of Person Signing Certificate

(2) Related Appeals and Interferences

There are no known related appeals and/or interferences.

(3) Status of Claims

Claims 1-4, 7-8, 10-12, 14, and 20-23 are pending in the case and stand rejected.

(4) Status of Amendments

No amendment has been filed after final rejection.

(5) Summary of Invention

The present patent application and claims define systems and techniques that allow client-generated activation requests to be fulfilled by a server, even if the client lacks information about any specific server that can process such requests. This allows client nodes to create remote components on available server nodes without monitoring the state of the network. (See the present specification at page 2, line 13 to page 3, line 3.) In an embodiment, the invention is implemented within a distributed component object model (DCOM) architecture. (See the present specification at FIG. 2 and page 5, line 17 to page 6, line 20.)

(6) Issues

Are claims 1-4, 7-8, 10-12, 14, and 20-23 improperly rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over U.S. Patent No. 6,226,684 (Sung) in view of the Background Section of the application as filed (Background)?

(7) Grouping of Claims

Claim 2 rises and falls with claim 1, and claim 12 rises and falls with claim 10, but otherwise, none of the claims rise and fall together for reasons set forth herein.

(8) Argument

Rejections under 35 USC § 103: Claims 1-4, 7-8, 10-12, 14, and 20-23 are improperly rejected based on Sung in view of the Background.

Claims 1-4, 7-8, 10-12, 14, and 20-23 stand rejected under 35 U.S.C. 103(a) as allegedly being unpatentable over Sung in view of the Background of the present application. This contention is respectfully traversed, and for reasons set forth herein, it is respectfully suggested that the rejection is improper because there is insufficient motivation to combine the references, and the proposed combination fails to teach or suggest all the claim limitations.

To establish a prima facie case of obviousness, three basic criteria must be met: (1) there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to combine reference teachings, (2) there must be a reasonable expectation of success, and (3) the combined prior art references must teach or suggest all the claim limitations. "The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure." See MPEP 706.02(j), citing In re Vaeck, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

The Background referred to is figure 1, and page 4, lines 3-10 of the present specification. These portions of applicant's disclosure describe a traditional distributed component object model architecture, where reusable program building blocks can be combined with one another in a distributed network to form an application, provided that identifiers for the reusable program building blocks, and a network name for a server to provide any remote program building blocks, are known (i.e., this information is present on the client at the time of program object creation).

The present claims define systems and techniques whereby a client can generate activation requests to be fulfilled by a server, even if the client lacks information about any specific server that can process such requests. This allows client nodes to create remote components on available server nodes without monitoring the state of the network, where the remote components comprise reusable program building blocks that are combinable with one or more other components in a distributed network to form an application.

In contrast, Sung does not describe object-oriented programming or a distributed component object model architecture. Rather, Sung teaches a method of "reestablishing connections between a particular client and a particular server by multiple routers." (See Sung at col. 13, lines 42-43.) Once a client is assigned to a particular server inside a server bank by a router, this assignment is saved for future use (in a "sticky" Internet Protocol (IP) cache table in the router), and this assignment of a client to a server is multicast to other routers in a collection of routers. Thus, when the client requests additional information from the server bank by sending a request to a router in the collection of routers at a later time, the same server in the server bank handles that information request, which can reduce excessive data caching in the server bank. (See Sung at Abstract; col. 1, line 7 to col. 2, line 32; and col. 5, lines 21-27.)

With respect to the first criteria necessary to establish a prima facie case of obviousness, no effective suggestion or motivation to combine Background with Sung has been identified, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art. The proposed motivation to combine is stated as follows: "because it would have an efficient communications system that can reuse or share a program with other components in the network servers." However, attempting to implement a distributed component object model architecture using Sung's techniques for reestablishing connections between a particular client and a particular server by using multiple routers, would in fact be a very inefficient approach, and hence, one which would <u>not</u> be selected by one having ordinary skill in the art for this reason.

Sung does not relate to a distributed component model architecture, and there is no indication of a reasonable expectation of success for the suggested combination. Adding multiple routers, each with their own sticky IP cache table, as taught by Sung, to a distributed component object model architecture would add <u>complexity and inefficiency</u> to the architecture without any apparent corresponding benefit. The clients in such a combination would still need to have stored a network address for at least one of the routers before a program object (which corresponds to a remote component provided by one of the servers) could be created.

Moreover, even if the hypothetical combination of Sung and Background were made, this combination would still not teach or suggest all the claim limitations.

Independent claim 1 requires a server node operating to enable the client node to activate remote components on available server nodes without specific names or capabilities of nodes in the network servicing the requests. Nothing in Sung in view of Background teaches or suggests this claimed feature of a server node. As described above, Sung teaches adding functionality to the routers, and not to the clients or servers. Thus, a prima facie showing of unpatentability has not been established for claim 1.

Dependent claim 3 specifies that the client node includes enhancements to a network protocol of the client node, and dependent claim 4 specifies that the server node includes enhancements to a network protocol of the server node. Nothing in Sung suggests that there are enhancements to a network protocol of either a client node or a server node, as claimed in this context. Rather, it is the routers in Sung that are enhanced. Thus, a prima facie showing of unpatentability has not been established for claim 3.

With respect to independent claim 23, the hypothetical combination would not result in client nodes configured to be able to request activation of remote components at run-time without specific names or capabilities of nodes servicing those requests, as claimed. If the proposed combination of Sung and Background could be made to enable a client node to activate remote components without specific names or capabilities of nodes servicing those requests, it would be the routers and not the servers or clients that would provide such flexibility, due to the very nature of the invention in Sung. Thus, a prima facie showing of unpatentability has not been established for claim 23.

With respect to independent claims 7, 14, 20 and 22, the proposed combination would not teach or suggest multicasting a machine-independent activation request to the network as claimed. The multicasting taught by Sung involves multicasting from one router to other routers information indicating the assignment of a client to a server. (See col. 2, lines 14-16; and col. 11, lines 31-42.) With respect to independent claims 10 and 21, the proposed combination would not teach or suggest monitoring, at a server, a specific port to receive a machine-independent client activation request. The server port referred to in Sung is used for all communications from the client. Thus, a prima facie showing of unpatentability has not been established for each of claims 7, 10, 14, 20, 21 and 22.

Dependent claim 8 specifies that the capability information includes a list of server IP (Internet Protocol) addresses or UNC (universal naming convention) names of servers that have the ability to service a request for a specific CLSID (Class Identifier). Sung neither teaches nor suggests sending of capability information and/or CLSIDs, which are globally unique identifiers used to refer to particular classes of objects in a distributed component object model architecture.

Dependent claim 11 specifies that monitoring the specific port at the server includes monitoring a port that is tied to a multicast IP address. In contrast, the port that is tied to a multicast address in Sung is monitored at the routers and not at the servers.

For all of these reasons, it is respectfully suggested that the rejection does not meet the patent office's burden of providing a prima facie showing of unpatentability. Thus, all pending claims, 1-4, 7-8, 10-12, 14, and 20-23, should be in condition for allowance.

The brief fee of $320 and $110 for a one month Petition of Extension of Time are enclosed. Please apply any other necessary charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: June 10, 2004

William E. Hunter
Reg. No. 47,671
Attorney for Intel Corporation

Fish & Richardson P.C.
Customer Number: 20985
4350 La Jolla Village Drive, Suite 500
San Diego, California 92122
Telephone: (858) 678-5070
Facsimile: (858) 678-5099

Appendix of Claims

1. A distributed component system in a network comprising:

a client node configured to process client activation requests; and

a server node configured to monitor activation requests from the client node, said node operating to enable the client node to activate remote components on available server nodes without specific names or capabilities of nodes in the network servicing the requests,

wherein said remote components comprise reusable program building blocks that are combinable with one or more other components in a distributed network to form an application.

2. The system of claim 1, wherein said network comprises a local-area network, a wide-area network, or Internet.

3. The system of claim 1, wherein said activation requests are processed by a client node that includes enhancements to a network protocol of the client node.

4. The system of claim 1, wherein said server node include enhancements to a network protocol of the server node.

5-6. (Cancelled)

7. A method comprising:

receiving a machine-independent activation request from a client in a network, wherein said activation request comprises a request to activate a reusable program building block that is combinable with one or more other reusable program building blocks in the network to form an application;

multicasting said activation request to the network; and

receiving capability information from servers available to service said activation request.

8. The method of claim 7, wherein the capability information includes a list of server IP addresses or UNC names of servers that have the ability to service a request for a specific CLSID.

9. (Cancelled)

10. A method comprising:

monitoring at a server a specific port to receive a machine-independent client activation request within a network, wherein said activation request comprises a request to activate a reusable program building block that is combinable with one or more other reusable program building blocks in the network to form an application;

retrieving a client address from an IP packet associated with the request; and

returning capability information of the server to the client address.

11. The method of claim 10, wherein monitoring the specific port includes monitoring a port that is tied to a multicast IP address.

12. The method of claim 10, wherein returning includes returning a server IP address.

13. (Cancelled)

14. A method comprising:

receiving a machine independent activation request from a client in a network, wherein said activation request comprises a request to activate a reusable program building block that is combinable with one or more other reusable program building blocks in the network to form an application;

multicasting said activation request to the network;

requesting capability information from servers available to service said activation request;

monitoring a port that is tied to a multicast IP address;

retrieving a client address from an IP packet; and

returning capability information of the server to the client address.

15-19. (Cancelled)

20. A computer program, residing on a computer readable medium, the program comprising executable instructions that enable the computer to:

receive a machine-independent activation request from a client in a network, wherein said activation request comprises a request to activate a reusable program building block that is combinable with one or more other reusable program building blocks in the network to form an application;

multicast said activation request to the network; and

receive capability information from servers available to service said activation request.

21. A computer program, residing on a computer readable medium, the program comprising executable instructions that enable the computer to:

monitor at a server a specific port that is tied to a multicast IP address to receive a machine-independent client activation request within a network, wherein said activation request comprises a request to activate a reusable program building block that is combinable with one or more other reusable program building blocks in the network to form an application;

retrieve a client address from an IP packet associated with the request; and

return capability information of the server to the client address.

22. A computer program, residing on a computer readable medium, the program comprising executable instructions that enable the computer to:

receive a machine-independent activation request from a client in a network, wherein said activation request comprises a request to activate a reusable program building block that is combinable with one or more other reusable program building blocks in the network to form an application;

multicast said activation request to the network;

request capability information from servers available to service said activation request;

monitor a port that is tied to a multicast IP address;

retrieve a client address from an IP packet; and

return capability information of the server to the client address.

23. A distributed component network comprising:

client nodes configured to be able to request activation of remote components at run-time without specific names or capabilities of nodes servicing those requests; and

server nodes operating to monitor the requests and respond appropriately to service the requests, wherein said remote components comprise reusable program building blocks that are combinable with one or more other components in the distributed component network to form an application.